

Como uma promise funciona

Aprendemos logo nos primeiros capítulos que as funções de `$http` (`get`, `post`, `put`, `delete`) devolvem uma promessa, no inglês *promise*. Uma promise contém o resultado futuro de uma ação. Quando a ação é bem sucedida, temos acesso ao valor retornado da ação, através da função `then` e erros através da função `catch`. Apesar de `$http` suportar as funções `success` e `error`, por baixo dos panos uma promise é utilizada.

Trabalhar com o *pattern* promise não é exclusividade do Angular, podemos implementá-lo em nosso código. Para isso, o Angular possui o serviço `$q`, especializado na criação de promises. Normalmente, retornamos uma promise a partir de uma chamada de função. Vejamos um exemplo:

```
function exhibe(texto) {
  return $q(function(resolve, reject) {

    // simulando ação assíncrona com setTimeout

    setTimeout(function() {
      if(texto == 'Alura') {
        resolve('resolvida');
      } else {
        reject('rejeitada')
      }
    }, 5000);
  });
}

// executando nossa promise

exib('Alura').then(function(resultado) {
  console.log(resultado);
}).catch(function(erro) {
  console.log(erro);
});
console.log('FIM');
```

O código acima é funcional, sendo assim, o que será exibido no console é:

Selecione uma alternativa

A Primeiro será exibido "resolvida" e depois "FIM"

B Primeiro será exibido "FIM" e depois "resolvida".

C Primeiro será exibido "rejeitada" e depois "FIM"

Primeiro será exibido "FIM" e depois "rejeitada"

D